

Санкт-Петербургский государственный университет

Прикладная математика и информатика
Нелинейная динамика, информатика и управление

Семак Дмитрий Андреевич

Самовозбуждающиеся и скрытые аттракторы в системе Хенона

Выпускная квалификационная работа

Научный руководитель:
к. ф.-м. н., в. н. с. Мокаев Т. Н.

Рецензент:
д. ф.-м. н., доцент Кузнецов Н. В.

Санкт-Петербург
2017

SAINT-PETERSBURG STATE UNIVERSITY

Applied Mathematics and Computer Science
Nonlinear dynamics, informatics and management

Semak Dmitry

Self-excited and hidden attractors in the Henon system

Bachelor's Thesis

Scientific supervisor:
Cand. Sc., Leading Researcher Timur Mokaev

Reviewer:
Dr. Sc., Associate Professor Nikolay Kuznetsov

Saint-Petersburg
2017

Оглавление

Введение	4
Динамические системы и аттракторы	9
1.1. Определение динамической системы	9
1.2. Аттракторы динамических систем	10
1.3. Скрытые и самовозбуждающиеся аттракторы	12
1.4. Способы локализации скрытых аттракторов	13
Локализация аттракторов	
в системе Хенона	16
2.1. Динамические свойства системы	16
2.2. Локализация скрытого аттрактора в дискретной системе Хенона	18
Заключение	27
Список литературы	28
Приложение А	32
Приложение В	49

Введение

Предположение о том, что кажущиеся случайными события возникают в системах, описывающих сложные процессы, появилось еще в середине 19-го века. Уже в 1860 году Джеймс Максвелл обсуждал, как столкновения молекул газа могут привести к постепенному усилению небольших изменений и вызвать микроскопическую хаотичность [29]. Позднее, в 1890 году, известный французский математик и физик Анри Пуанкаре обнаружил чувствительную зависимость от начальных условий решения системы, описывающей взаимодействие трех тел в частном случае, и после этого сделал предположение, что такие явления могут быть вполне обычными [30].

В общем использовании хаос понимается как полное отсутствие связи между случайными событиями. С научной точки зрения понятие хаоса связано с явным отсутствием порядка в системе, которая, тем не менее, подчиняется определенным законам или правилам. Хаос в этом смысле является синонимом динамической нестабильности. Это свойство связано с присущим хаосу отсутствием предсказуемости в некоторых физических системах. Фундаментальные работы Анри Пуанкаре, а также последующие работы Алексея Михайловича Ляпунова, Джорджа Дейвида Биркгофа и Андрея Николаевича Колмогорова легли в основу теории хаоса, а также дали толчок к развитию нелинейной динамики и качественной теории дифференциальных уравнений.

Однако до некоторого момента времени считалось, что хаос в системе возникает из-за ее сложности, а некоторые системы и вовсе не поддавались стандартному анализу. Так, например, уравнения Навье-Стокса для потока жидкости в общем случае невозможно проанализировать традиционными математическими методами, но определенные упрощенные дифференциальные уравнения, которые описывали частные случаи в потоке жидкости, могли поддаваться анализу и демонстрировать явление хаоса. Так, в 1944 году была опубликована статья советского физика Льва Давидовича Ландау «К проблеме турбулентности» [35]. В своей работе он анализировал периодические и квази-

периодические уравнения и предположил, что турбулентность в жидкости возникает в результате каскада последовательных *бифуркаций* — изменений характера движения динамической системы на большом временном промежутке при изменении одного или нескольких параметров системы. Аналогичное предположение развивал и немецкий математик Эберхард Хопф в своей работе 1948 года [21], поэтому данный подход к возникновению турбулентности еще называют сценарием Ландау-Хопфа.

В 1971 году Давид Руэль и Флорис Такенс подвергли критике исследования Ландау. По их мнению, достаточно было рассмотрения небольшого числа частот, чтобы динамика системы стала турбулентной. Такое турбулентное поведение системы было связано с появлением в фазовом пространстве нового типа аттракторов — *странного аттрактора*. Он также являлся притягивающим множеством в фазовом пространстве системы, но траектории на нем вели себя неустойчиво, а его геометрическая структура представляла собой то, что в дальнейшем стало называться фрактальным множеством или фракталом. Таким образом, появление в этой системе странного аттрактора стало признаком начала турбулентности и положило начало нового подхода к исследованию динамических систем.

В 1963 году американский математик и метеоролог Эдвард Лоренц опубликовал статью, в которой он исследовал процесс конвекции жидкости в двумерном слое и обнаружил более сложное поведение системы [28]. Анализ этого поведения был тесно связан с феноменом хаоса, а именно чувствительной зависимости от начальных условий, а последующее моделирование на ЭВМ подтвердило хаотическое поведение системы и позволило визуализировать первый странный хаотический аттрактор в фазовом пространстве системы. Впоследствии данная трехмерная система получила название «система Лоренца», и до сих пор она имеет огромный научный интерес.

Так, в 1969 году системой Лоренца заинтересовался французский математик и астроном Мишель Хенон (Энон или Эно) [13]. В своей работе он показал, что важные и существенные свойства динамических

систем, определяемые дифференциальными уравнениями, могут быть сохранены с помощью тщательно подобранных приближающих отображений. За основу была взята трехмерная система Лоренца, а в качестве приближающего отображения — сечение Пуанкаре [4], что в результате дало отображение, которое сейчас нам известно как отображение Хенона (или Henon map). Оно задано следующим образом:

$$\begin{aligned}x &\rightarrow 1 - ax^2 + y \\y &\rightarrow bx\end{aligned}\tag{1.1}$$

Стоит отметить, что ранее отображение Хенона рассматривалось строго как математическое преобразование плоскости. Однако есть и физические интерпретации, одна из которых была выявлена Офером Бихамом и Вольфгангом Венцелем [3]. Они показали, что траектории отображения однозначно отождествляются с экстремумами определенной функции Гамильтона, описывающей бесконечную цепочку взаимно взаимодействующих атомов под воздействием внешнего потенциала. Также из последних исследований стало известно, что отображение Хенона эквивалентно отображению периодически колеблющегося гармонического осциллятора с подходящей нелинейной связью [12], поэтому это отображение имеет не только математический интерес, но интерес с точки зрения физики.

Отображение Хенона, как и система Лоренца, имеет в своем фазовом пространстве странный хаотический аттрактор. Он возникает из-за последовательности бифуркаций удвоения периода цикла или сценария Фейгенбаума [11]. Однако странный аттрактор существует не при всех параметрах системы, из-за чего возникает задача его численной локализации.

С вычислительной точки зрения важную роль в исследовании систем играет вопрос локализации предельных множеств или аттракторов в фазовом пространстве. Аттрактор представляет собой множество точек в фазовом пространстве системы, к которому приближаются траектории после переходного процесса, а его притягивающее множество называется *бассейном притяжения*. В начальный период развития тео-

рии нелинейной динамики и теории хаоса больше внимания уделялось анализу систем, для которых проблема возникновения периодических и хаотических колебаний решалась довольно просто. Это происходило из-за того, что структура большинства динамических систем приводила к появлению *самовозбуждающихся колебаний* и процессов, которые появлялись из окрестности неустойчивых состояний равновесия. Например, численно обнаруженные хаотические колебания в системе Лоренца, а также последующие найденные колебания в известных системах Рёслера, Чуа и др. оказались самовозбуждающимися. Чтобы обнаружить такие колебания, используется *стандартная вычислительная процедура*, в которой траектории строятся из точки неустойчивого многообразия в окрестности неустойчивого состояния равновесия и притягиваются к некоторому аттрактору, тем самым задавая его.

Однако последующие исследования показали, что существует еще один тип аттракторов, которые позднее были названы *скрытыми аттракторами*. В отличие от самовозбуждающихся аттракторов, бассейн притяжения скрытых аттракторов не пересекается с окрестностями неустойчивых состояний равновесия, что усложняет их численную локализацию. Появление такого типа аттракторов относят ко второй части 16-ой проблемы Гильберта о взаимном расположении предельных циклов и их количестве в двумерных системах [20]. Также скрытые аттракторы возникали в гипотезах Айзермана и Кальмана [22, 34], и являлись контрпримерами к их предположениям. В 2010 году Геннадий Александрович Леонов и Николай Владимирович Кузнецов предложили новый метод поиска скрытых аттракторов в многомерных системах [26], который в дальнейшем позволил впервые обнаружить скрытый хаотический аттрактор в электронном контуре Чуа [23, 24].

В работе [10] скрытые аттракторы в системе Хенона были локализованы с помощью *нестационарных точек с нулевым ускорением* (или *perpetual points*). Предположение, что скрытые аттракторы могут быть локализованы из такого типа точек, ранее было выдвинуто в работах [31, 32], но это предположение до сих пор не доказано аналитически.

Основной целью данной работы является локализация скрытых

аттракторов в фазовом пространстве системы Хенона, посредством *процедуры продолжения по параметру* и верификация результатов, полученных в работе [10].

Динамические системы и аттракторы

1.1. Определение динамической системы

Основываясь на работах [5, 27, 36], введем основные понятия теории динамических систем. Пусть (M, ρ) — метрическое пространство, введем множество моментов времени $\mathbb{T} \in \{\mathbb{R}, \mathbb{R}_+, \mathbb{Z}, \mathbb{Z}_+\}$.

Определение 1. Пусть $\{\phi^t\}_{t \in \mathbb{T}}$ — семейство отображений, $\phi^t : M \rightarrow M$. Пара

$$(\{\phi^t\}_{t \in \mathbb{T}}, (M, \rho)) \quad (1.2)$$

есть *динамическая система* на метрическом пространстве (M, ρ) , если выполнены условия:

1. $\phi^0 = id_M$ — тождественное отображение на M ;
2. $\phi^{t+s} = \phi^t \circ \phi^s$ для любых моментов времени $s, t \in \mathbb{T}$;
3. Если $\mathbb{T} \in \{\mathbb{R}, \mathbb{R}_+\}$, то $\phi^{(\cdot)}(\cdot) : \mathbb{T} \times M \rightarrow M$ — непрерывное отображение; если же $\mathbb{T} \in \{\mathbb{Z}, \mathbb{Z}_+\}$, то для любого $t \in \mathbb{T}$ $\phi^t(\cdot) : M \rightarrow M$ является непрерывным отображением.

Пространство (M, ρ) называется *фазовым пространством* динамической системы.

Если $\mathbb{T} \in \{\mathbb{R}, \mathbb{R}_+\}$, то динамическая система называется *системой с непрерывным временем или потоком*. Если $\mathbb{T} \in \{\mathbb{Z}, \mathbb{Z}_+\}$, то динамическая система называется *системой с дискретным временем*.

Определение 2. Пусть задана динамическая система (1.2). Для произвольного $p \in M$ множество $\psi(p) := \{\phi^t(p) \mid t \in \mathbb{T}\}$ есть *траектория* динамической системы через точку p . Если $\mathbb{T} \in \{\mathbb{R}, \mathbb{Z}\}$, то можно рассматривать *положительную* $\psi^+(p) := \{\phi^t(p) \mid t \in \mathbb{T}_+\}$ и *отрицательную полутраекторию* $\psi^-(p) := \{\phi^t(p) \mid t \in \mathbb{T}_-\}$. Движение $t \in \mathbb{T} \rightarrow \phi^t(p)$ называется *стационарным*, если

$$\forall t \in \mathbb{T} \quad \phi^t(p) = p.$$

Движение $t \in \mathbb{T} \rightarrow \phi^t(p)$ называется *периодическим*, если оно не стационарно и существует момент времени $T \in \mathbb{T}_+$ такой, что для любого $t \in \mathbb{T}$ имеем $\phi^{t+T}(p) = \phi^t(p)$, где число T — *период* движения.

Траектория стационарного движения есть *состояние равновесия*. Траектория периодического движения называется *циклом* или *периодической траекторией*.

Отображение Хенона (1.1) удовлетворяет всем трем свойствам из определения (Опр. 1) и порождает динамическую систему с дискретным временем ($\mathbb{T} = \mathbb{Z}_+$, $M = \mathbb{R}^2$).

1.2. Аттракторы динамических систем

Одним из важнейших определений в теории динамических систем является понятие аттрактора, которое возникает при изучении предельного поведения траекторий.

Определение 3. Будем называть множество $Z \subset M$ *инвариантным* относительно динамической системы (1.2), если для любого $t \in \mathbb{T}$ имеет место $\phi^t(Z) = Z$. При этом множество Z называется *положительно инвариантным*, если для любого $t \in \mathbb{T}_+$ имеем $\phi^t(Z) \subset Z$, и *отрицательно инвариантным*, если для любого $t \in \mathbb{T}_+$ имеем $Z \subset \phi^t(Z)$.

Определение 4. Множество $Z \subset M$ называется *притягивающим* (\mathfrak{B} - *притягивающим*) для множества $W \subset M$ относительно динамической системы (1.2), если

$$\lim_{t \rightarrow \infty} \rho(\phi^t(p), Z) = 0, \quad \forall p \in W,$$

$$(\lim_{t \rightarrow \infty} \rho(\phi^t(B), Z) = 0, \forall B \subset W, \quad B — \text{ограниченное множество})$$

где $\rho(\phi^t(p), Z)$ расстояние от точки $\phi^t(p)$ до множества Z , определяемое по формуле

$$\rho(\phi^t(p), Z) = \inf_{q \in Z} \|\phi^t(p) - q\|.$$

Свойство 1. Будем говорить, что множество Z является *глобально притягивающим* (\mathfrak{B} - *глобально притягивающим*) относительно динамической системы (1.2), если $W = M$ и Z — притягивающее (\mathfrak{B} - притягивающее).

Определение 5. Будем называть множество Z *аттрактором* (\mathfrak{B} - *аттрактором*) для W относительно (1.2), если оно обладает свойствами:

1. Z — ограничено и замкнуто;
2. Z — инвариантное множество;
3. Z является притягивающим для W относительно (1.2) (Z является \mathfrak{B} -притягивающим для W относительно (1.2)). Так же предполагается, что $\text{int } W \neq \emptyset$.

Наибольшее множество W , обладающее свойствами 1 – 3, называется областью притяжения Z .

Замечание 1. Если множество Z — \mathfrak{B} притягивающее для W относительно (1.2), то Z — притягивающее для W относительно (1.2). Обратное не верно.

Замечание 2. Стоит отметить, что иногда в определении аттрактора свойство замкнутости опускается (см. [2]). Аналогично иногда опускается свойство ограниченности (см. [6]). Из первого замечания следует, что глобальный \mathfrak{B} аттрактор также является глобальным аттрактором. Поэтому необходимо ввести понятие *минимального глобального аттрактора* (и *минимального аттрактора*), то есть множества Z , обладающего свойствами:

1. ограниченное, замкнутое и минимальное;
2. положительно инвариантное;
3. глобально притягивающее множество (притягивающее множество).

В дальнейшем под *аттрактором* (и *глобальным аттрактором*) будет пониматься *минимальный аттрактор* (и *минимальный глобальный аттрактор*).

Определение 6. Для аттрактора Z бассейном притяжения является множество начальных состояний $B(Z) \subset M$, такое что

$$\lim_{t \rightarrow \infty} \rho(Z, \phi^t(p)) = 0, \quad \forall p \in B(Z).$$

1.3. Скрытые и самовозбуждающиеся аттракторы

В настоящее время теория нелинейных динамических систем является одной из наиболее исследуемых областей математики, которая используется во многих отраслях, например в энергетике, промышленности, авиации и других. Именно поэтому основное внимание в исследованиях направлено на анализ устойчивости состояний равновесия системы и возникновение колебаний в ней. При этом структура многих исследуемых систем была такой, что колебания возникали из неустойчивых состояний равновесия, поэтому существование такого рода колебаний в системах было практически очевидным. Локализовать предельные множества таких колебаний позволяла стандартная вычислительная процедура.

С вычислительной точки зрения такое колебание или возможные конечные состояния системы получили названия аттрактора, а множество начальных данных, для которых траектории стремятся к аттрактору — бассейном притяжения. Именно эти понятия являются ключевыми определениями в теории динамических систем. Исходя из этого, следует ввести следующую классификацию аттракторов.

Определение 7. Будем называть аттрактор *самовозбуждающимся*, если бассейн притяжения данного аттрактора пересекается с любыми окрестностями его стационарных точек, в противном случае аттрактор будет называться *скрытым*.

Проблема локализации скрытых колебаний появилась еще в 1900 году в 16-й проблеме Гильберта для двумерных полиномиальных си-

стем [20]. До сих пор эта проблема остается актуальной, хотя было получено множество численных и теоретических результатов. Задача анализа и локализации скрытых колебаний возникает и в прикладных отраслях. Так, в конце прошлого века проблема возникла при моделировании систем управления угловым положением летальных аппаратов и при моделировании буровых установок [9, 14, 18] [1, 7, 8, 15–17, 19, 25, 33]. Наличие скрытых колебаний в этих системах может привести к неожиданным устойчивым и неустойчивым решениям, которые могут привести к поломке или разрушению оборудования. В 2010 году был впервые рассчитан и смоделирован хаотический скрытый аттрактор в обобщенном контуре Чуа [23], а в последствии и в классическом контуре Чуа [24], что являлось огромным прорывом в области исследования скрытых колебаний.

1.4. Способы локализации скрытых аттракторов

Задача численной локализации скрытых аттракторов достаточно трудоемка и требует разработку и применение более сложных методов. В данной работе рассматриваются два метода решения этой задачи. Первый метод основывается на предположении о том, что скрытые аттракторы численно локализуются из окрестности *нестационарных точек с нулевым ускорением (или perpetual points)*. По мнению авторов предложенной концепции [10, 31, 32], эти точки служат своеобразным ориентиром для поиска скрытых аттракторов, также как неустойчивые состояния равновесия для самовозбуждающихся аттракторов. Дадим более формальное определение данных точек.

Определение 8. Пусть задана динамическая система (1.2). Точка $p \in M$ называется *нестационарной точкой с нулевым ускорением*, если

$$\frac{d\phi^t(p)}{dt} \neq 0 \quad \forall t \in \mathbb{T},$$

$$\frac{d^2\phi^t(p)}{dt^2} = 0 \quad \forall t \in \mathbb{T}.$$

Если система с дискретным временем $\mathbb{T} \in \{\mathbb{Z}, \mathbb{Z}_+\}$, то можно аппроксимировать первые и вторые производные, и условия переписутся следующим образом

$$D\phi^t = \frac{\phi^{t+1}(p) - \phi^t(p)}{t+1-t} \neq 0 \quad \forall t \in \mathbb{T},$$

$$D^2\phi^t = \frac{D\phi^{t+1}(p) - D\phi^t(p)}{t+1-t} = 0 \quad \forall t \in \mathbb{T}.$$

Однако с помощью данного метода не всегда можно локализовать скрытый аттрактор, например, при определенных параметрах системы нестационарные точки с нулевым ускорением могут не существовать.

Второй способ основан на *теории гомотопий* и *процедуре продолжения по параметру*. В нем строится последовательность подобных друг другу систем так, чтобы для начальной системы предельное множество траекторий могло быть получено аналитически. Так, в качестве начальной системы используют систему с самовозбуждающимся аттрактором. Рассмотрим динамическую систему

$$(\{\phi_\lambda^t\}_{t \in \mathbb{T}, \lambda \in \Lambda}, M), \quad (1.3)$$

с параметром $\lambda \in \Lambda$, где $\Lambda \subset \mathbb{R}^d$ – открытое множество и $M \subset \mathbb{R}^n$ – открытое (M – фазовое пространство, Λ – пространство параметров). Определим $\lambda_{end} \in \Lambda$ как точку, в которой осуществляется поиск скрытого аттрактора. Выберем точку $\lambda_{begin} \in \Lambda$ такую, что можно численно или аналитически локализовать некоторый нетривиальный аттрактор \mathcal{A}^1 системы (1.3). Временной интервал $t \in [0, T]$ выбирается достаточно большим. В пространстве параметров определим *путь* $\gamma : [0, 1] \rightarrow \Lambda$ такой, что $\gamma(0) = \lambda_{begin}$ и $\gamma(1) = \lambda_{end}$. Так же введем последовательность точек $\{\lambda_j\}_{j=1}^k$ на пути, где $\lambda_1 = \lambda_{begin}$, $\lambda_k = \lambda_{end}$ и дистанция между λ_j и λ_{j+1} достаточно мала. На каждом шаге j процедуры начальная точка траектории $\psi_j(0)$ выбирается так, чтобы $\psi_j(0) = \psi_{j-1}(T)$. Следуя этому условию и увеличивая j , возможны два случая:

- Точки аттрактора \mathcal{A}^j лежат в бассейне притяжения аттрактора \mathcal{A}^{j+1} ;

- При переходе от системы с λ_j к системе λ_{j+1} наблюдается потеря устойчивости бифуркаций, и аттрактор \mathcal{A}^j исчезает.

Если при изменениях λ_j от λ_{begin} до λ_{end} не наблюдается потери устойчивости бифуркаций, и аттрактор сохраняется, тогда в конце процедуры при $\lambda_k = \lambda_{end}$ произойдет локализация скрытого аттрактора.

Метод продолжения по параметру позволяет локализовать скрытые аттракторы в тех случаях, где не будут существовать точки с нулевым ускорением, но стоит отметить, что этот метод достаточно трудоемкий с вычислительной точки зрения, особенно при большом количестве параметров. Именно поэтому в сложных системах предварительно проводится дополнительное исследование и анализ системы.

Локализация аттракторов в системе Хенона

2.1. Динамические свойства системы

Проанализируем динамику системы (1.1) в фазовой плоскости, состоящей из двух переменных (x, y) , при этом меняя значения a и b . Параметр b характеризует степень *диссипативности* отображения, то есть насколько уменьшится малый элемент фазового пространства при увеличении числа итераций отображения. Именно в таких системах возможно существование аттракторов. Обычно параметр b рассматривается на промежутке $[0, 1]$. Параметр a — параметр нелинейной части отображения (1.1). Найдем неподвижные точки системы из условия:

$$x = 1 - ax^2 + y$$

$$y = bx.$$

При условии

$$a > (1 - b)^2/4$$

существуют два действительных решения, которые задают координаты неподвижных точек в фазовом пространстве:

$$x_1 = \frac{-(1 - b) - \sqrt{(1 - b)^2 + 4a}}{2a},$$

$$y_1 = bx_1,$$

$$x_2 = \frac{-(1 - b) + \sqrt{(1 - b)^2 + 4a}}{2a},$$

$$y_2 = bx_2.$$

Собственные значения матрицы Якоби находятся из уравнения:

$$\det \begin{bmatrix} -2ax^* - \lambda & 1 \\ b & -\lambda \end{bmatrix} = 0.$$

В результате получатся следующие собственные числа:

$$(x_1, y_1) : \lambda_{1,2} = \left\{ \frac{1}{2}(1 - b + c \mp \sqrt{4c + (1 + b - c)^2}) \right\},$$

и

$$(x_2, y_2) : \lambda_{3,4} = \left\{ \frac{1}{2}(1 - b - c \mp \sqrt{-4c + (1 + b + c)^2}) \right\},$$

где $c = \sqrt{4a + (b - 1)^2}$.

Если $a \rightarrow 0$, то неподвижные точки уходят на бесконечность. Если зафиксировать $b > 0$, то точка с координатами (x_1, y_1) является седлом при любых a , так как собственные значения матрицы Якоби в этой точке $\lambda_2 > 0$, $\lambda_1 < 0$. Точка с координатами (x_2, y_2) является устойчивым узлом при $b = 0.3$, так как оба собственных значения матрицы Якоби в этой точке отрицательны. При $a = 0.3675$ одно из собственных значений, соответствующее точке (x_2, y_2) , принимает значение -1 , и происходит бифуркация удвоения периода и рождение цикла периода 2.

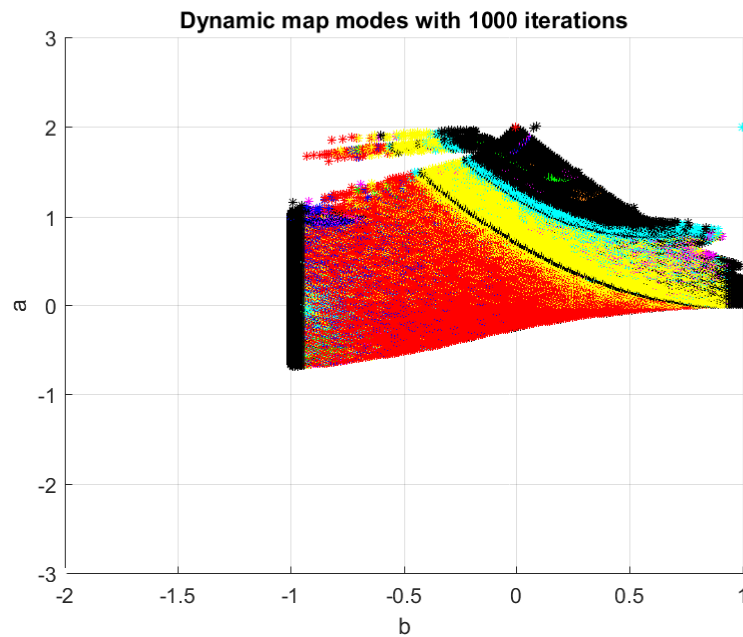
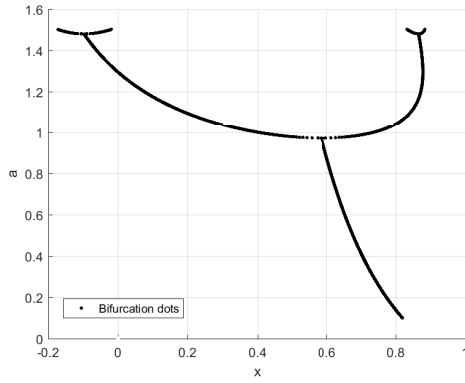
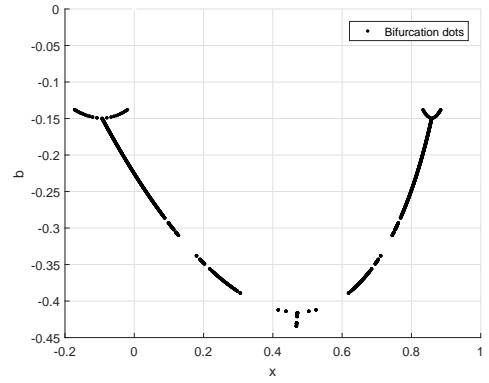


Рис. 2.1: Карта динамических режимов системы (1.1): цикл периода 1 - красный; цикл периода 2 - желтый; цикл периода 3 - синий; цикл периода 4 - бирюзовый; цикл периода 5 - зеленый; цикл периода 6 - розовый; цикл периода 7 - оранжевый; хаос или циклы периода больше 7 - черный.



(a) Бифуркационная диаграмма удвоения периода цикла для параметра a .



(b) Бифуркационная диаграмма удвоения периода цикла для параметра b .

Рис. 2.2: Бифуркационные диаграммы системы (1.1).

При увеличении параметра a цикл периода 2 переходит в цикл периода 4 и так далее. Вследствие этого происходит переход к хаосу в системе (1.1). Наглядно такое поведение отражает карта динамических режимов (см. рис. 2.1) и бифуркационные диаграммы (см. рис. 2.2). Таким образом, переход к хаосу в системе (1.1) происходит через каскад бифуркаций удвоения периода цикла, который называется сценарием Фейгенбаума [11].

2.2. Локализация скрытого аттрактора в дискретной системе Хенона

Как было показано в предыдущем разделе, система (1.1) имеет два состояния равновесия:

$$FP1 = \left(\frac{-(1-b) - \sqrt{(1-b)^2 + 4a}}{2a}, \frac{-b((1-b) - \sqrt{(1-b)^2 + 4a})}{2a} \right),$$

$$FP2 = \left(\frac{-(1-b) + \sqrt{(1-b)^2 + 4a}}{2a}, \frac{-b((1-b) + \sqrt{(1-b)^2 + 4a})}{2a} \right).$$

При фиксированных параметрах

$$a = 1.8, \quad b = -0.05$$

состояние равновесия $FP1$ является устойчивым узлом, а состояние равновесия $FP2$ является неустойчивым узлом, так как собственные значения соответствующие этим точкам равны

$$FP1 : -1.8037, -0.0277,$$

$$FP2 : 3.9186, 0.0128.$$

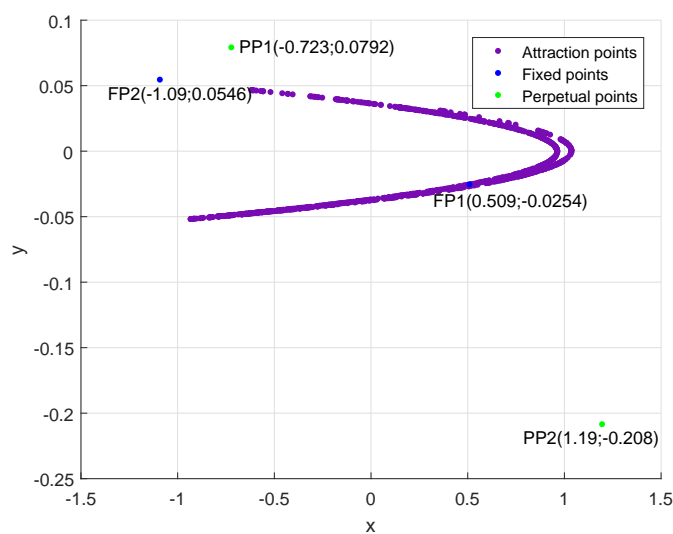
В фазовом пространстве при таких параметрах существует самовозбуждающийся аттрактор. С помощью стандартной вычислительной процедуры его легко визуализировать (см. рис. 2.3).

Оба состояния равновесия $FP1$ и $FP2$, обозначенные синими точками, попадают в бассейн притяжения (см. рис. 2.3b) этого аттрактора, следовательно, данный аттрактор действительно является самовозбуждающимся.

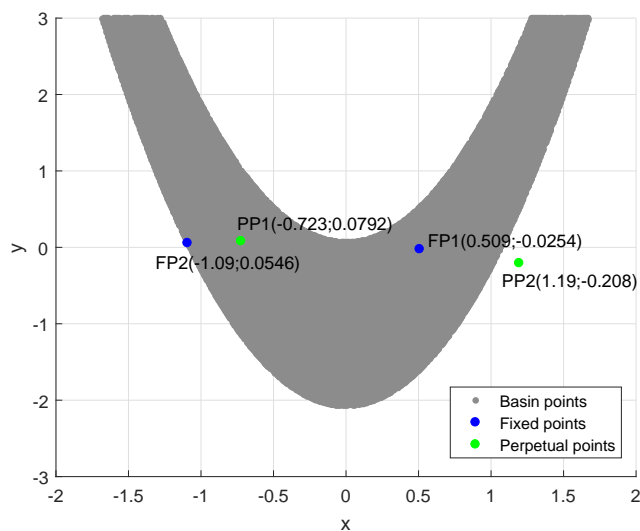
В результатах работы [10] было обнаружено, что скрытый аттрактор в системе (1.1) был локализован из нестационарной точки с нулевым ускорением при параметрах $a = 1.49$ и $b = -0.138$. Используя определение этих точек (Опр. 8) и определение системы (1.1), а также то, что это система с дискретным временем, перепишем неравенства из определения (Опр. 8) следующим образом:

$$\begin{cases} 1 - ax^2 + y - x \neq 0; \\ bx - y \neq 0; \\ 1 - a(1 - ax^2 + y)^2 + bx - 2(1 - ax^2 + y) + x = 0; \\ b(1 - ax^2 + y) - 2bx + y = 0. \end{cases}$$

Решениями данной системы будут являться нестационарные точки с нулевым ускорением, представленные ниже:



(a) Самовозбуждающийся аттрактор (фиолетовый), построенный в окрестности состояния равновесия $FP2$.



(b) Бассейн притяжения самовозбуждающегося аттрактора.

Рис. 2.3: Самовозбуждающийся аттрактор системы (1.1) при $a = 1.8$, $b = -0.05$ и его бассейн притяжения .

- при $a \neq 0$ и $b\sqrt{4a + 5b^2 + 6b + 5} - 2b^2 - 2b \neq 0$

$$x_{per} = \frac{-\sqrt{4a + 5b^2 + 6b + 5} + 3b + 1}{2a},$$

$$y_{per} = \frac{b(-3\sqrt{4a + 5b^2 + 6b + 5} + 7b + 5)}{2a}.$$

- при $a \neq 0$ и $-b\sqrt{4a + 5b^2 + 6b + 5} - 2b^2 - 2b \neq 0$

$$x_{per} = \frac{\sqrt{4a + 5b^2 + 6b + 5} + 3b + 1}{2a},$$

$$y_{per} = \frac{b(3\sqrt{4a + 5b^2 + 6b + 5} + 7b + 5)}{2a}.$$

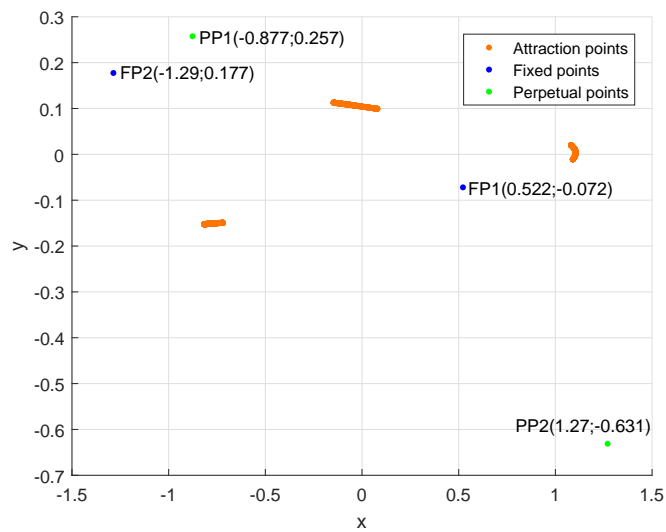
- при $a = 0$ и $b = 1$

$$y_{per} = \frac{1}{2}(2x_{per} - 1).$$

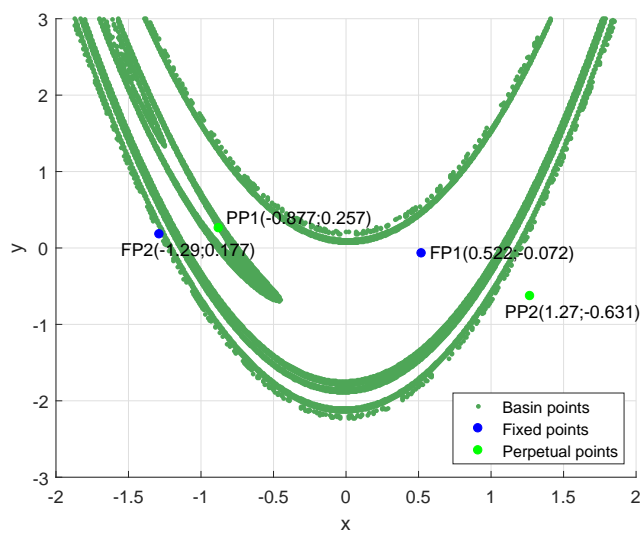
Взяв параметры $a = 1.49$ и $b = -0.138$ из работы [10], были получены аттракторы из окрестностей нестационарных точек с нулевым ускорением, обозначаемые $PP1$ и $PP2$ соответственно. Из окрестности точки $PP1 = (-0.877, 0.257)$ удалось локализовать скрытый аттрактор (см. рис. 2.4), полученный в статье [10]. От второй точки $PP2 = (1.27, -0.631)$ локализовать аттрактор не удалось.

С помощью дополнительного моделирования было обнаружено, что бассейн притяжения аттрактора, построенного из нестационарной точки с нулевым ускорением при параметрах $a = 1.49$ и $b = -0.138$, и самовозбуждающийся аттрактор, построенный при параметрах $a = 1.8$ и $b = -0.05$ пересекаются (см. рис. 2.5). Следовательно, при помощи алгоритма продолжения по параметру за одну итерацию можно перейти от пары параметров $a = 1.8$ и $b = -0.05$ к паре $a = 1.49$ и $b = -0.138$ и получить скрытый аттрактор, изображенный на рисунке 2.4.

Однако попытки локализовать скрытый аттрактор, полученный из окрестности точки с нулевым ускорением $PP1$ (см. рис. 2.4) с помощью процедуры продолжения по параметру за большее число шагов, не увенчались успехом, так как в фазовом пространстве при параметрах $a = 1.49$ и $b = -0.138$ существует еще один аттрактор — самовозбужда-



(a) Скрытый аттрактор (оранжевый) системы (1.1), полученный из точки с нулевым ускорением $PP1$.



(b) Бассейн притяжения скрытого аттрактора.

Рис. 2.4: Скрытый аттрактор и его бассейн притяжения при $a = 1.49$ и $b = -0.138$.

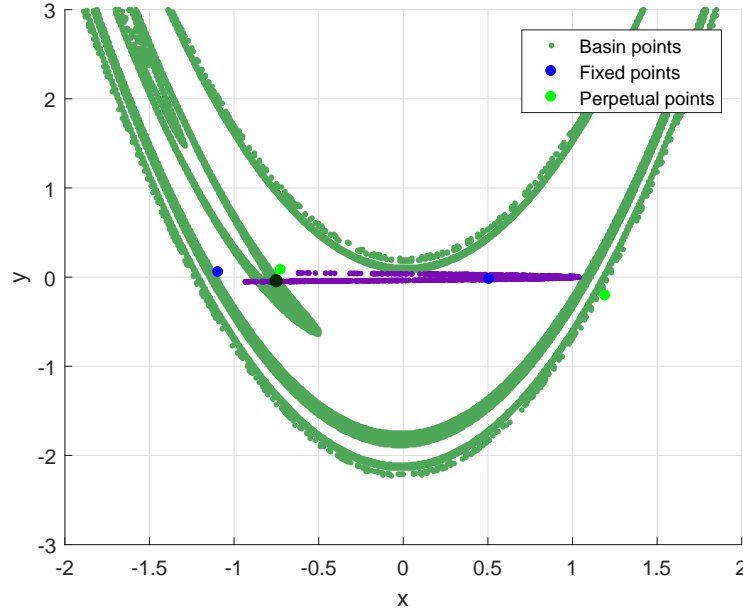


Рис. 2.5: Пересечение бассейна притяжения аттрактора, полученного из окрестности точки с нулевым ускорением $PP1 = (-0.877, 0.257)$ при $a = 1.49$ и $b = -0.138$ и самовозбуждающегося аттрактора при $a = 1.8$ и $b = -0.05$ в точке $(-0.740261; -0.049043)$ (черная точка).

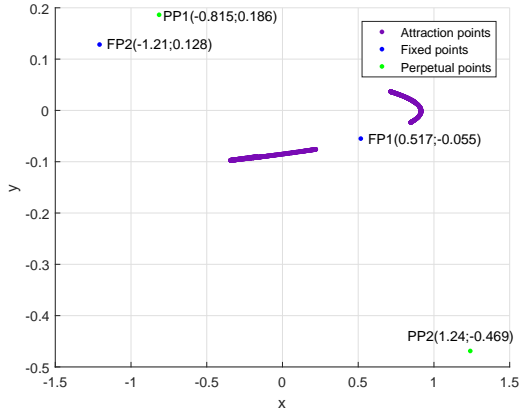
ющийся устойчивый цикл периода 4. В процессе работы алгоритма, начиная с некоторого шага, траектории выпущенные из последней точки траектории, построенной на предыдущем шаге, притягиваются к этому циклу (см. рис. 2.6) и на всех последующих итерациях притягиваются только к нему.

Также дополнительное моделирование показало, что скрытый аттрактор изображенный на рисунке 2.7а и в статье [10] на самом деле является самовозбуждающимся аттрактором. При параметрах $a = 1.49$ и $b = -0.138$ система (1.1) имеет два состояния равновесия: $FP1 = (0.522, -0.072)$ и $FP2 = (-1.29, 0.177)$. Собственные значения матрицы Якоби, соответствующие данным точкам:

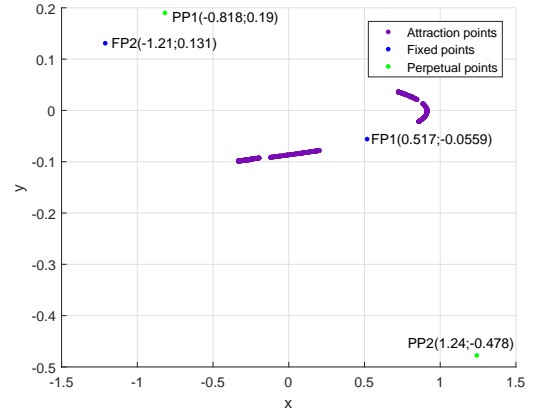
$$FP1 : -1.4611, -0.0945,$$

$$FP2 : 3.7952, 0.0364.$$

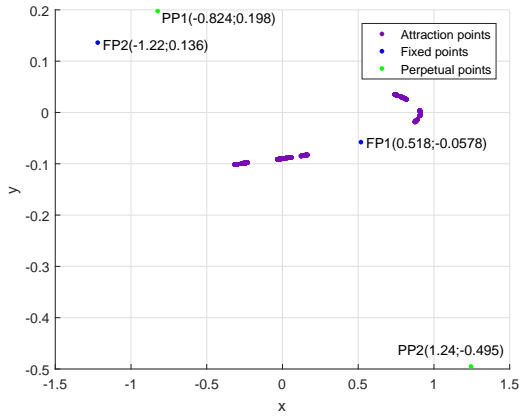
Таким образом, состояние равновесия $FP1$ является устойчивым узлом, а $FP2$ является неустойчивым узлом. При проверке окрестностей ста-



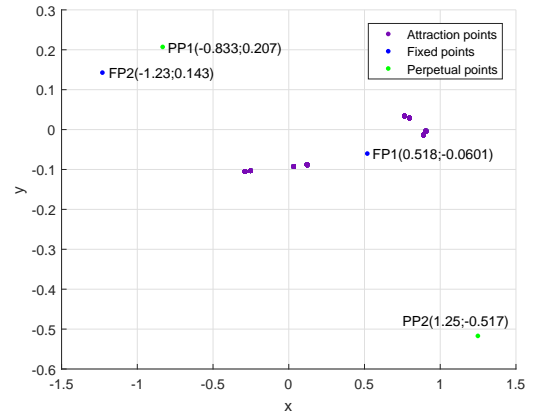
(a) Итерация 63, $a = 1.60160$,
 $b = -0.10632$.



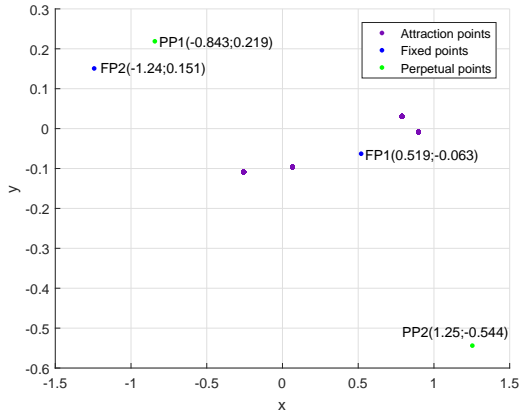
(b) Итерация 66, $a = 1.59540$,
 $b = -0.10808$.



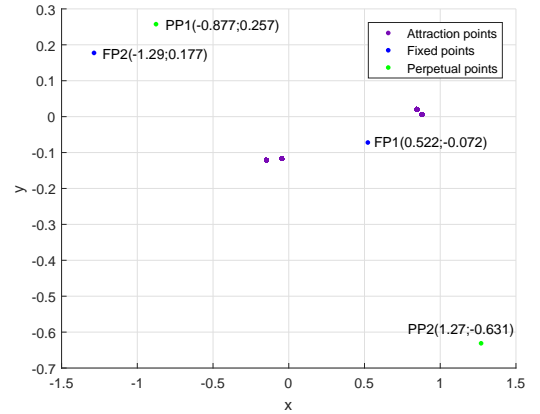
(c) Итерация 69, $a = 1.58300$,
 $b = -0.11160$.



(d) Итерация 75, $a = 1.56750$,
 $b = -0.11600$.



(e) Итерация 81, $a = 1.54890$,
 $b = -0.12128$.



(f) Итерация 100, $a = 1.49000$,
 $b = -0.13800$.

Рис. 2.6: Часть последовательности преобразований в процедуре продолжения по параметру для системы (1.1), начиная с пары параметров (1.8; -0.05) до (1.49; -0.138).

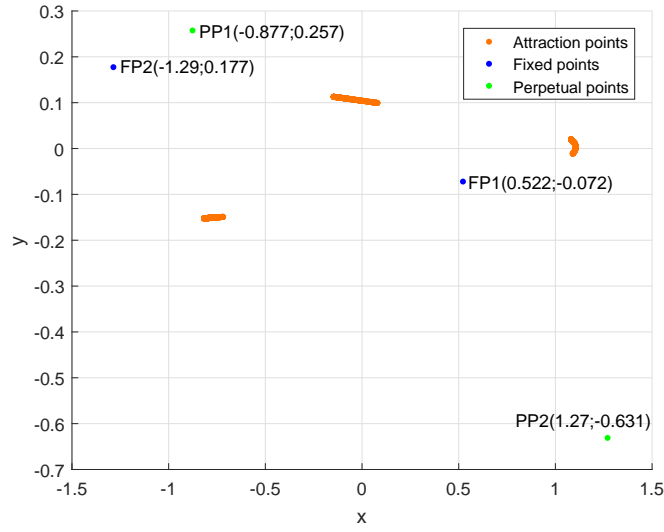
ционарной точки $FP2$ было обнаружено, что из окрестности неустойчивого многообразия размера 10^{-9} удается локализовать скрытый аттрактор, полученный из нестационарной точки с нулевым ускорением (см. рис 2.7). Следовательно, бассейн изображенный на рисунке 2.4b и окрестность точки $FP2$ пересекаются, из чего можно сделать вывод, что этот аттрактор является хаотическим самовозбуждающимся, а не скрытым. Что противоречит результатам работы [10].

Дополнительно были вычислены параметры a и b , при которых существует либо одна из двух нестационарных точек с нулевым ускорением, либо не существует ни одной. Ниже приведены условия существования:

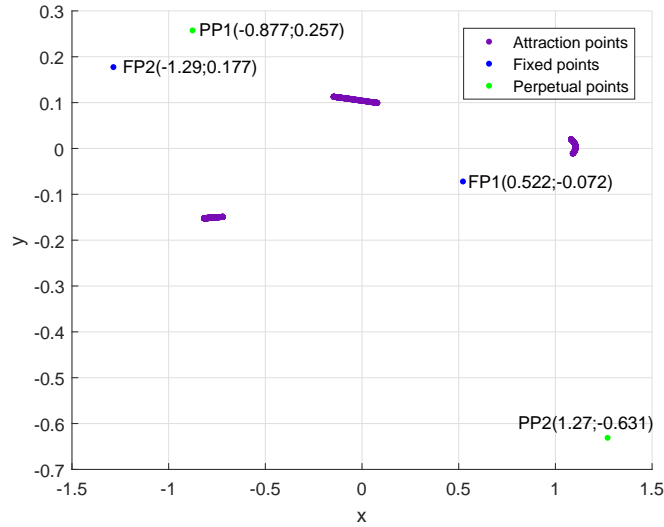
- нестационарные точки с нулевым ускорением не существуют, когда либо $a = 0$, либо $b = 0$, либо $\sqrt{(4a + 5b^2 + 6b + 5)} + 2b + 2 = 0$ и $\sqrt{(4a + 5b^2 + 6b + 5)} - 2b - 2 = 0$;
- существует хотя бы одна нестационарная точка с нулевым ускорением, когда $a \neq 0$, $b \neq 0$ и либо $\sqrt{(4a + 5b^2 + 6b + 5)} + 2b + 2 \neq 0$, либо $\sqrt{(4a + 5b^2 + 6b + 5)} - 2b - 2 \neq 0$.

Однако локализовать скрытые аттракторы с помощью процедуры продолжения по параметру в этих случаях не удалось.

В рамках исследования было проверено множество пар параметров: параметр $b \in [-0.1, -0.9]$ и параметр $a \in [0.1, 1.9]$. В процессе работы алгоритма рассматривались окрестности точек размера от 10^{-1} до 10^{-10} , чтобы более точно определить тип аттрактора, но возникали ситуации при которых даже этого набора окрестностей было недостаточно, как и в случае скрытого аттрактора из работы [10]. Это связано с тем, что система (1.1) дискретна и траектории, выходящие из неустойчивого многообразия состояния равновесия, являются точками, а не непрерывными линиями, поэтому очень важно рассмотреть все случаи, чтобы максимально точно определить тип аттрактора.



(a) Скрытый аттрактор (оранжевый) системы (1.1), полученный из точки с нулевым ускорением $PP1$ из статьи [10].



(b) Самовозбуждающийся аттрактор (фиолетовый), полученный из окрестности неустойчивого многообразия размера 10^{-9} .

Рис. 2.7: Скрытый и самовозбуждающийся аттрактор при $a = 1.49$ и $b = -0.138$. Самовозбуждающийся аттрактор получен из начальной точки $x_0 = (-1.285744592288457; 0.177432753837378)$.

Заключение

В данной работе были решены следующие задачи:

1. Реализован алгоритм построения бифуркаций, бассейнов притяжения и карты динамических режимов для системы Хенона;
2. Реализован алгоритм локализации скрытых аттракторов при помощи нестационарных точек с нулевым ускорением;
3. Разработан алгоритм локализации скрытых аттракторов посредством процедуры продолжения по параметру;
4. Проведена численная верификация гипотезы и результатов, полученных в статье [10], показывающая, что результаты не верны.

Список литературы

- [1] Aircraft control with anti-windup compensation / G.A. Leonov, B.R. Andrievskii, N.V. Kuznetsov, A.Yu. Pogromskii // Differential equations. — 2012. — Vol. 48, no. 13. — P. 1700–1720.
- [2] Babin A. V. Global attractors in PDE // Handbook of Dynamical Systems. — Elsevier Science, 2006. — Vol. 1B. — P. 983–1085.
- [3] Biham O., Wenzel W. Characterization of unstable periodic orbits in chaotic attractors and repellers // Phys. Rev. Lett. — 1989. — Vol. 63. — P. 819.
- [4] Birkhoff G. D. Dynamical systems with two degrees of freedom // Trans. Amer. Math. Soc. — 1917. — Vol. 18. — P. 199–300.
- [5] Boichenko V. A., Leonov G. A., Reitmann V. Dimension Theory for Ordinary Differential Equations. — Stuttgart : Teubner, 2005.
- [6] Chepyzhov V. V., Goritskii A. Yu. Unbounded attractors of evolution equations // Adv. Sov. Math. — 1992. — Vol. 10. — P. 85–128.
- [7] Convergence based anti-windup design method and its application to flight control / B.R. Andrievsky, N.V. Kuznetsov, G.A. Leonov, A.Yu. Pogromsky // International Congress on Ultra Modern Telecommunications and Control Systems and Workshops. — IEEE, 2012. — P. 212–218 (art. no. 6459667).
- [8] Discontinuity and Complexity in Nonlinear Physical Systems / R.E. Best, N.V. Kuznetsov, G.A. Leonov et al. — Springer, 2014. — Vol. 6.
- [9] Drilling systems failures and hidden oscillations / M. A. Kiseleva, N. V. Kuznetsov, G. A. Leonov, P. Neittaanmäki // IEEE 4th International Conference on Nonlinear Science and Complexity, NSC 2012 - Proceedings. — 2012. — P. 109–112.

- [10] Dudkowski D., Prasad A., Kapitaniak T. Perpetual points and hidden attractors in dynamical systems // *Physics Letters A*. — 2015. — Vol. 379, no. 40-41. — P. 2591 – 2596.
- [11] Feigenbaum M. J. The universal metric properties of nonlinear transformations // *J. Stat. Phys.* — 1979. — Vol. 21, no. 6. — P. 669–706.
- [12] Heagy J. F. A physical interpretation of the Hénon map // *Physica D Nonlinear Phenomena*. — 1992. — Vol. 57. — P. 436–446.
- [13] Henon M. A two-dimensional mapping with a strange attractor // *Communications in Mathematical Physics*. — 1976. — Vol. 50, no. 1. — P. 69–77.
- [14] Hidden Oscillations in Aircraft Flight Control System with Input Saturation / B. R. Andrievsky, N. V. Kuznetsov, G. A. Leonov, A. Yu. Pogromsky // *IFAC Proceedings Volumes*. — 2013. — Vol. 46, no. 12. — P. 75–79.
- [15] Hidden oscillations in drilling system actuated by induction motor / M.A. Kiseleva, N.V. Kuznetsov, G.A. Leonov, P. Neittaanmaki // *IFAC Proceedings Volumes (IFAC-PapersOnline)*. — 2013. — Vol. 5. — P. 86–89.
- [16] Hidden oscillations in drilling systems: torsional vibrations / G.A. Leonov, M.A. Kiseleva, N.V. Kuznetsov, P. Neittaanmäki // *Journal of Applied Nonlinear Dynamics*. — 2013. — Vol. 2, no. 1. — P. 83–94.
- [17] Hidden oscillations in drilling systems with salient pole synchronous motor / M.A. Kiseleva, N.V. Kondratyeva, N.V. Kuznetsov, G.A. Leonov // *IFAC-PapersOnLine*. — 2015. — Vol. 48, no. 11. — P. 700–705.
- [18] Hidden oscillations in mathematical model of drilling system actuated by induction motor with a wound rotor / G. A. Leonov,

- N. V. Kuznetsov, M. A. Kiseleva et al. // Nonlinear Dynamics. — 2014. — Vol. 77, no. 1-2. — P. 277–288.
- [19] Hidden periodic oscillations in drilling system driven by induction motor / M.A. Kiseleva, N.V. Kondratyeva, N.V. Kuznetsov et al. // IFAC Proceedings Volumes. — 2014. — Vol. 47, no. 19. — P. 5872–5877.
 - [20] Hilbert D. Mathematical problems // Bulletin of the American Mathematical Society. — 1902. — Vol. 8. — P. 437–479.
 - [21] Hopf E. A mathematical example displaying features of turbulence // Comm. Appl. Math. — 1948. — Vol. 1. — P. 303–322.
 - [22] Kalman R. E. Physical and Mathematical mechanisms of instability in nonlinear automatic control systems // Transactions of ASME. — 1957. — Vol. 79, no. 3. — P. 553–566.
 - [23] Kuznetsov N.V., Leonov G.A., Vagaitsev V.I. Analytical-numerical method for attractor localization of generalized Chua’s system // IFAC Proceedings Volumes. — 2010. — Vol. 43, no. 11. — P. 29–33.
 - [24] Kuznetsov N.V., Leonov G.A., Vagaitsev V.I. Localization of hidden Chua’s attractors // Physics Letters A. — 2011. — Vol. 375, no. 23. — P. 2230–2233.
 - [25] Leonov G.A., Kuznetsov N.V., Pogromskii A.Yu. Stability domain analysis of an antiwindup control system for an unstable object // Doklady Mathematics. — 2012. — Vol. 86, no. 1. — P. 587–590.
 - [26] Leonov G. A. Effective methods for periodic oscillations search in dynamical systems // App. math. & mech. — 2010. — Vol. 74, no. 1. — P. 24–50.
 - [27] Leonov G. A., Mokaev T. N. Lyapunov Dimension Formula for the Attractor of the Glukhovsky-Dolzhangsky System // Doklady Mathematics. — 2016. — Vol. 93, no. 1. — P. 42–45.

- [28] Lorenz E. N. Deterministic nonperiodic flow // J. Atmos. Sci. — 1963. — Vol. 20, no. 2. — P. 130–141.
- [29] Maxwell J. On the Dynamical Theory of Gases // Philosophical Transactions of the Royal Society of London. — 1867. — Vol. 157. — P. 49–88.
- [30] Poincaré H. Sur le problème des trois corps et les équations de la dynamique // Acta Mathematica. — 1890. — Vol. 13. — P. 1–270.
- [31] Prasad A. Existence of perpetual points in nonlinear dynamical systems and its applications // International Journal of Bifurcation and Chaos. — 2015. — Vol. 25, no. 2. — art. num. 1530005.
- [32] Prasad A. A Note On Topological Conjugacy For Perpetual Points // International Journal of Nonlinear Science. — 2016. — Vol. 21, no. 1. — P. 60–64.
- [33] van den Berg Roel, Pogromsky Alexander, Rooda Koos. Convergent Systems Design: Anti-Windup for Marginally Stable Plants // 45th IEEE Conference on Decision and Control. — Optical Society of America, 2006. — P. 5441–5446.
- [34] Айзерман М. А. Об одной проблеме, касающейся устойчивости в.
- [35] Ландау Л. Д. К проблеме турбулентности // Доклады Академии наук. — 1944. — Vol. 44. — P. 339.
- [36] Ф. Райтманн. Динамические системы, аттракторы и оценки их размерности. — Издательство Санкт-Петербургского университета, 2013.

Приложение А

Реализация алгоритма локализации скрытых аттракторов в системе Хенона

Листинг 1: `henonFixedPoints.m` — вычисление стационарных точек системы Хенона.

```
1 function [FP outFlag] = henonFixedPoints(a, b, number,
   outTextFlag)
2     FP1(1) = (-(1 - b) + sqrt((1 - b)^2 + 4 * a))/(2 * a);
3     FP1(2) = b * FP1(1);
4     FP2(1) = (-(1 - b) - sqrt((1 - b)^2 + 4 * a))/(2 * a);
5     FP2(2) = b * FP2(1);
6     outFlag = true;
7     if (number == 1)
8         if (imag(FP1(1)) ~= 0 || imag(FP1(2)) ~= 0)
9             if (outTextFlag)
10                 fprintf ('FP1 complex point, the algorithm is not
   running!');
11             end
12             FP = -1;
13             outFlag = false;
14             return;
15         end
16         if (isinf(FP1(1)) || isinf(FP1(2)))
17             if (outTextFlag)
18                 fprintf ('FP1 Inf point, the algorithm is not
   running!');
19             end
20             FP = -1;
21             outFlag = false;
22             return;
23         end
24         if (isnan(FP1(1)) || isnan(FP1(2)))
25             if (outTextFlag)
26                 fprintf ('FP1 NaN point, the algorithm is not
   running!');
27             end
```



```

28         FP = -1;
29         outFlag = false;
30         return;
31     end
32 end
33 if (number == 2)
34     if (imag(FP2(1)) ~= 0 || imag(FP2(2)) ~= 0)
35         if (outTextFlag)
36             fprintf ('FP1 complex point, the algorithm is not
37 running!');
38         end
39         FP = -1;
40         outFlag = false;
41         return;
42     end
43     if (isinf(FP2(1)) || isinf(FP2(2)))
44         if (outTextFlag)
45             fprintf ('FP1 Inf point, the algorithm is not
46 running!');
47         end
48         FP = -1;
49         outFlag = false;
50         return;
51     end
52     if (isnan(FP2(1)) || isnan(FP2(2)))
53         if (outTextFlag)
54             fprintf ('FP1 NaN point, the algorithm is not
55 running!');
56         end
57         FP = -1;
58         outFlag = false;
59         return;
60     end
61 end
62 if (number == 1)
63     FP = FP1;
64 end
65 if (number == 2)
66     FP = FP2;
67 end

```

65 **end**

Листинг 2: **henonPerpetualPoints.m** — вычисление нестационарных точек с нулевым ускорением системы Хенона.

```
1 function PPs = henonPerpetualPoints(a, b)
2     PPs = zeros(2, 2);
3     PPs(1:end, 1:end) = sqrt(-1.0);
4     if (a ~= 0) && ((b*sqrt(4*a+5*b^2+6*b+5)-2*b^2-2*b) ~= 0)
5         PPs(1, 1) = (-sqrt(4*a+5*b^2+6*b+5)+3*b+1)/(2*a);
6         PPs(1, 2) = b*(-3*sqrt(4*a+5*b^2+6*b+5)+7*b+5)/(2*a);
7     end
8     if (a ~= 0) && ((-b*sqrt(4*a+5*b^2+6*b+5)-2*b^2-2*b) ~= 0)
9         PPs(2, 1) = (sqrt(4*a+5*b^2+6*b+5)+3*b+1)/(2*a);
10        PPs(2, 2) = b*(3*sqrt(4*a+5*b^2+6*b+5)+7*b+5)/(2*a);
11    end
12    end
```

Листинг 3: **henonVicinityPoints.m** — вычисление стартовых точек из окрестности неустойчивого многообразия стационарных точек.

```
1 function [x0 flag] = henonVicinityPoints(a, b, S0, delta)
2     function out = J( x, a, b )
3         out = [-2 * a * x(1), 1; b, 0];
4     end
5     [V0, D0] = eig(J([S0(1), S0(2)], a, b));
6     [~, IX0] = sort(diag(D0), 'descend');
7     if (D0(IX0(1), IX0(1)) > 0 )
8         x0 = zeros(2, 2);
9         x0(1, :) = [S0(1), S0(2)] - delta * V0(:, IX0(1))'/norm(
10            V0(:, IX0(1)));
11        x0(2, :) = [S0(1), S0(2)] + delta * V0(:, IX0(1))'/norm(
12            V0(:, IX0(1)));
13        flag = true;
14    else
15        x0 = -1;
16        flag = false;
17    end
18    end
```

Листинг 4: **checkInitialPoint.m** — процедура проверки стартовой точки на комплексность, бесконечность и неопределенность.

```
1 function [x0 outFlag] = checkInitialPoint(a, b, FP, line, delta,
    outTextFlag)
2     if (isnan(FP(1, 1)) == 0) && (isnan(FP(1, 2)) == 0) && (isinf
    (FP(1, 1)) == 0) && (isinf(FP(1, 2)) == 0)
3         [x2 flag] = henonVicinityPoints(a, b, FP, delta);
4     else
5         if (outTextFlag)
6             disp('It is impossible to calculate initial point!')
7         ;
8         end
9         x0 = - 1;
10        outFlag = false;
11        return;
12    end
13    if (~flag)
14        if (outTextFlag)
15            disp('initial point stable point!');
16        end
17        x0 = -1;
18        outFlag = false;
19        return;
20    end
21    if (imag(x2(line, 1)) ~= 0 || imag(x2(line, 2)) ~= 0)
22        if (outTextFlag)
23            disp('initial point complex point!');
24        end
25        x0 = -1;
26        outFlag = false;
27        return;
28    end
29    x0 = x2(line, :);
30    outFlag = true;
end
```

Листинг 5: **runHenonCM.m** — запуск процедуры продолжения по параметру для системы Хенона.

```
1 function runHenonCM
```

```

2     clc
3     outDirPathS11 = './OUT/Way from SelfExcited to Hidden
attraction/S11';
4     outDirPathS12 = './OUT/Way from SelfExcited to Hidden
attraction/S12';
5     outDirPathS21 = './OUT/Way from SelfExcited to Hidden
attraction/S21';
6     outDirPathS22 = './OUT/Way from SelfExcited to Hidden
attraction/S22';
7     if ~exist(outDirPathS11, 'dir')
8         mkdir(outDirPathS11);
9     end
10    if ~exist(outDirPathS12, 'dir')
11        mkdir(outDirPathS12);
12    end
13    if ~exist(outDirPathS21, 'dir')
14        mkdir(outDirPathS21);
15    end
16    if ~exist(outDirPathS22, 'dir')
17        mkdir(outDirPathS22);
18    end
19    aBegin = 1.8; aEnd = 1.49;
20    bBegin = -0.05; bEnd = -0.138;
21    % aBegin = 1.43; aEnd = 1.49;
22    % bBegin = 0.02; bEnd = -0.138;
23    % aBegin = 1.7; aEnd = 1.538;
24    % bBegin = 0.1; bEnd = -0.101;
25    % aBegin = 1.7; aEnd = 1.6;
26    % bBegin = -0.13; bEnd = -0.13;
27    numSteps = 100;
28    iterations = 1000;
29    a = aBegin;
30    b = bBegin;
31    stepSize = 1/numSteps;
32    delta = 10-3;
33    [FP1 outFlag1] = henonFixedPoints(a, b, 1, true);
34    if (outFlag1)
35        [x0 outFlag2] = checkInitialPoint(a, b, FP1, 1, delta,
true);
36        if (outFlag2)

```

```

37         searchHiddenAttraction(a, b, aEnd, bEnd, numSteps,
searchSize, iterations, x0, outDirPathS11);
38     end
39 end
40 [FP1 outFlag1] = henonFixedPoints(a, b, 1, true);
41 if (outFlag1)
42     [x0 outFlag2] = checkInitialPoint(a, b, FP1, 2, delta,
true);
43     if (outFlag2)
44         searchHiddenAttraction(a, b, aEnd, bEnd, numSteps,
searchSize, iterations, x0, outDirPathS12);
45     end
46 end
47 [FP2 outFlag1] = henonFixedPoints(a, b, 2, true);
48 if (outFlag1)
49     [x0 outFlag2] = checkInitialPoint(a, b, FP2, 1, delta,
true);
50     if (outFlag2)
51         searchHiddenAttraction(a, b, aEnd, bEnd, numSteps,
searchSize, iterations, x0, outDirPathS21);
52     end
53 end
54 [FP2 outFlag1] = henonFixedPoints(a, b, 2, true);
55 if (outFlag1)
56     [x0 outFlag2] = checkInitialPoint(a, b, FP2, 2, delta,
true);
57     if (outFlag2)
58         searchHiddenAttraction(a, b, aEnd, bEnd, numSteps,
searchSize, iterations, x0, outDirPathS22);
59     end
60 end

```

Листинг 6: **searchHiddenAttraction.m** — процедура продолжения по параметру для системы Хенона.

```

1 function searchHiddenAttraction (aBegin, bBegin, aEnd, bEnd,
numSteps, stepSize, iterations, x0, outDir)
2     outDir = [outDir '/' '(' num2str(aBegin, '%.6g') ';' num2str(
bBegin, '%.6g') ')') ' to (' num2str(aEnd, '%.6g') ';' num2str(
bEnd, '%.6g') ')')'];

```

```

3     outDirAttractFromPP = [outDir '/Attractors from perpetual
point'];
4     outDirHiddens = [outDir '/Hidden attractors'];
5     outDirInterAttr = [outDir '/Intermediate attractors'];
6     if ~exist(outDir, 'dir')
7         mkdir(outDir);
8     else
9         rmdir(outDir, 's');
10        mkdir(outDir);
11    end
12    if ~exist(outDirAttractFromPP, 'dir')
13        mkdir(outDirAttractFromPP);
14    else
15        rmdir(outDirAttractFromPP, 's');
16        mkdir(outDirAttractFromPP);
17    end
18    if ~exist(outDirHiddens, 'dir')
19        mkdir(outDirHiddens);
20    else
21        rmdir(outDirHiddens, 's');
22        mkdir(outDirHiddens);
23    end
24    if ~exist(outDirInterAttr, 'dir')
25        mkdir(outDirInterAttr);
26    else
27        rmdir(outDirInterAttr, 's');
28        mkdir(outDirInterAttr);
29    end
30    disp('Starting building way from SelfExcited to Hidden
attractor...');
31    a = aBegin;
32    b = bBegin;
33    eps = 1e-3;
34    stepSizeBasin = 0.01;
35    i = 0;
36    while (i <= numSteps)
37        fprintf('Iteration = %f a = %.15f b = %.15f\n', i, a, b)
;
38        attractorDots = buildingHenonAttractor(a, b, iterations,
x0, '', '', '', '', false);

```

```

39         if (checkSelfExcited(a, b, iterations, attractorDots))
40             name = ['/ ' num2str(i) ' - Self-Excited attractor ('
num2str(a, '%.6g') '; ' num2str(b, '%.6g') ').fig'];
41             line1 = sprintf('Self-Excited attractor with %.0f
iterations', iterations);
42             line2 = sprintf('a = %.5f, b = %.5f', a, b);
43             color = [121 11 181]./255;
44             inTitle = {line1, line2};
45             buildingHenonAttractor(a, b, iterations, x0, name,
outDirInterAttr, inTitle, color, true);
46         end
47         PPs = henonPerpetualPoints(a, b);
48         PP(1) = PPs(1, 1); PP(2) = PPs(1, 2);
49         x1(1, 1) = PP(1); x1(1, 2) = PP(2);
50         attractorPP1Dots = buildingHenonAttractor(a, b,
iterations, x1, '', '', '', false);
51         if (~checkSelfExcited(a, b, iterations, attractorPP1Dots)
)
52             line1 = sprintf('Hidden attractor from perpetual
point with %.0f iterations', iterations);
53             line2 = sprintf('a = %.5f, b = %.5f', a, b);
54             color = [255 115 0]./255;
55             inTitle = {line1, line2};
56             name = ['/ ' num2str(i) ' - Hidden attractor from PP1
(' num2str(a, '%.6g') '; ' num2str(b, '%.6g') ').fig'];
57             buildingHenonAttractor(a, b, iterations, x1, name,
outDirAttractFromPP, inTitle, color, true);
58         else
59             line1 = sprintf('Self-Excited attractor from
perpetual point with %.0f iterations', iterations);
60             line2 = sprintf('a = %.5f, b = %.5f', a, b);
61             color = [12 171 126]./255;
62             inTitle = {line1, line2};
63             name = ['/ ' num2str(i) ' - Self-Excited attractor PP1
(' num2str(a, '%.6g') '; ' num2str(b, '%.6g') ').fig'];
64             buildingHenonAttractor(a, b, iterations, x1, name,
outDirAttractFromPP, inTitle, color, true);
65         end
66         PP(1) = PPs(2, 1); PP(2) = PPs(2, 2);
67         x1(1, 1) = PP(1); x1(1, 2) = PP(2);

```

```

68     attractorPP2Dots = buildingHenonAttractor(a, b,
iterations, x1, '', '', '', false);
69     if (~checkSelfExcited(a, b, iterations, attractorPP2Dots)
)
70         line1 = sprintf('Hidden attractor from perpetual
point with %.0f iterations', iterations);
71         line2 = sprintf('a = %.5f, b = %.5f', a, b);
72         color = [255 115 0]./255;
73         inTitle = {line1, line2};
74         name = ['/ ' num2str(i) ' - Hidden attractor from PP2
(' num2str(a, '%.6g') ' ; ' num2str(b, '%.6g') ') .fig'];
75         buildingHenonAttractor(a, b, iterations, x1, name,
outDirAttractFromPP, inTitle, color, true);
76     else
77         line1 = sprintf('Self-Excited attractor from
perpetual point with %.0f iterations', iterations);
78         line2 = sprintf('a = %.5f, b = %.5f', a, b);
79         color = [12 171 126]./255;
80         inTitle = {line1, line2};
81         name = ['/ ' num2str(i) ' - Self-Excited attractor PP2
(' num2str(a, '%.6g') ' ; ' num2str(b, '%.6g') ') .fig'];
82         buildingHenonAttractor(a, b, iterations, x1, name,
outDirAttractFromPP, inTitle, color, true);
83     end
84     a = aBegin + (i + 1) * stepSize * ( aEnd - aBegin );
85     b = vectorSyst(a, aBegin, aEnd, bBegin, bEnd);
86     x0(1, 1) = attractorDots(1, end);    x0(1, 2) =
attractorDots(2, end);
87     name = ['/ ' num2str(i) ' - Next Self-Excited attractor ( '
num2str(a, '%.6g') ' ; ' num2str(b, '%.6g') ') .fig'];
88     line1 = sprintf('Self-Excited attractor with %.0f
iterations', iterations);
89     line2 = sprintf('a = %.5f, b = %.5f', a, b);
90     color = [0 0 0]./255;
91     inTitle = {line1, line2};
92     attractorDotsNext = buildingHenonAttractor(a, b,
iterations, x0, name, outDirInterAttr, inTitle, color, true);
93     name = ['/ ' num2str(i) ' - Basin next self-excited
attractor for parameter ( ' num2str(a, '%.6g') ' ; ' num2str(b, '
%.6g') ') .fig'];

```



```

94     basin = basinAttractionHenonSyst(a, b, -3, 3, -3, 3,
stepSizeBasin, iterations, attractorDotsNext, name,
outDirInterAttr, true);
95     if (any(ismembertol(basin(1, :), attractorDots(1, :), eps
)) && any(ismembertol(basin(2, :), attractorDots(2, :), eps)))
96         if (checkSelfExcited(a, b, iterations,
attractorDotsNext))
97             i = i + 1;
98             x0(1, 1) = attractorDots(1,end);    x0(1, 2) =
attractorDots(2,end);
99             a = aBegin + i * stepSize * ( aEnd - aBegin );
100             b = vectorSyst(a, aBegin, aEnd, bBegin, bEnd);
101             continue;
102         else
103             fprintf (' ! Found Hidden attractor a = %.15f b =
%.15f x0 = (%.15f; %.15f)\n', a, b, x0(1), x0(2));
104             name = ['/ ' num2str(i) ' - Hidden attractor ('
num2str(a, '%.6g') ' ; ' num2str(b, '%.6g') ') .fig'];
105             line1 = sprintf('Hidden attractor with %.0f
iterations', iterations);
106             line2 = sprintf('a = %.5f, b = %.5f', a, b);
107             color = [237 143 20]./255;
108             inTitle = {line1, line2};
109             attractorHiddenDots = buildingHenonAttractor(a, b
, iterations, x0, name, outDirHiddens, inTitle, color, true);
110             name = ['/ ' num2str(i) ' - Basin hidden attractor
for parameter (' num2str(a, '%.6g') ' ; ' num2str(b, '%.6g') ')
.fig'];
111             basinAttractionHenonSyst(a, b, -3, 3, -3, 3,
stepSizeBasin, iterations, attractorHiddenDots, name,
outDirHiddens, true);
112             PPs = henonPerpetualPoints(a, b);
113             PP(1) = PPs(1, 1); PP(2) = PPs(1, 2);
114             x1(1, 1) = PP(1); x1(1, 2) = PP(2);
115             attractorPP1Dots = buildingHenonAttractor(a, b,
iterations, x1, '', '', '', false);
116             if (~checkSelfExcited(a, b, iterations,
attractorPP1Dots))
117                 line1 = sprintf('Hidden attractor from
perpetual point with %.0f iterations', iterations);

```

```

118         line2 = sprintf('a = %.5f, b = %.5f', a, b);
119         color = [255 115 0]./255;
120         inTitle = {line1, line2};
121         name = ['/ ' num2str(i) ' - Hidden attractor
from PP1 (' num2str(a, '%.6g') ';' num2str(b, '%.6g') ').fig'
];
122         buildingHenonAttractor(a, b, iterations, x1,
name, outDirAttractFromPP, inTitle, color, true);
123     else
124         line1 = sprintf('Self-Excited attractor from
perpetual point with %.0f iterations', iterations);
125         line2 = sprintf('a = %.5f, b = %.5f', a, b);
126         color = [12 171 126]./255;
127         inTitle = {line1, line2};
128         name = ['/ ' num2str(i) ' - Self-Excited
attractor from PP1 (' num2str(a, '%.6g') ';' num2str(b, '%.6g')
) ').fig'];
129         buildingHenonAttractor(a, b, iterations, x1,
name, outDirAttractFromPP, inTitle, color, true);
130     end
131     PP(1) = PPs(2, 1); PP(2) = PPs(2, 2);
132     x1(1, 1) = PP(1); x1(1, 2) = PP(2);
133     attractorPP2Dots = buildingHenonAttractor(a, b,
iterations, x1, '', '', '', false);
134     if (~checkSelfExcited(a, b, iterations,
attractorPP2Dots))
135         line1 = sprintf('Hidden attractor from
perpetual point with %.0f iterations', iterations);
136         line2 = sprintf('a = %.5f, b = %.5f', a, b);
137         color = [255 115 0]./255;
138         inTitle = {line1, line2};
139         name = ['/ ' num2str(i) ' - Hidden attractor
from PP2 (' num2str(a, '%.6g') ';' num2str(b, '%.6g') ').fig'
];
140         buildingHenonAttractor(a, b, iterations, x1,
name, outDirAttractFromPP, inTitle, color, true);
141     else
142         line1 = sprintf('Self-Excited attractor from
perpetual point with %.0f iterations', iterations);
143         line2 = sprintf('a = %.5f, b = %.5f', a, b);

```

```

144         color = [12 171 126]./255;
145         inTitle = {line1, line2};
146         name = ['/ ' num2str(i) ' — Self-Excited
attractor from PP2 ( ' num2str(a, '%.6g') ' ; ' num2str(b, '%.6g'
) ').fig'];
147         buildingHenonAttractor(a, b, iterations, x1,
name, outDirAttractFromPP, inTitle, color, true);
148     end
149 end
150 break;
151 else
152     fprintf ('Basin and attractor not have common points,
algorithm stoped!\n');
153     break;
154 end
155 end
156 end

```

Листинг 7: **checkSelfExcited.m** — процедура определения типа аттрактора.

```

1 function out = checkSelfExcited(a, b, iterations, attractionDots)
2     n = 0;
3     numberDelta = 10;
4     eps = 1e-6;
5     for i = 1:2
6         for j = 1:2
7             for l = 1:numberDelta
8                 flag1(i, j, l) = -1;
9                 delta(l) = 10^(-l);
10            end
11        end
12    end
13    [FP1 outFlag1] = henonFixedPoints(a, b, 1, false);
14    if (outFlag1)
15        n = n + 1;
16        xCurr(n) = FP1(1);
17        yCurr(n) = FP1(2);
18    end
19    [FP2 outFlag2] = henonFixedPoints(a, b, 2, false);

```

```

20     if (outFlag2)
21         n = n + 1;
22         xCurr(n) = FP2(1);
23         yCurr(n) = FP2(2);
24     end
25     if (n ~= 0)
26         for i = 1:n
27             FP(1, 1) = xCurr(i); FP(1, 2) = yCurr(i);
28             for p = 1:numberDelta
29                 [x0 outFlag3] = checkInitialPoint(a, b, FP, 1,
delta(p), false);
30                 if (outFlag3)
31                     %fprintf('%d1 x0 = (%.10f; %.10f)\n', p, x0
(1, 1), x0(1, 2));
32                     attraction = buildingHenonAttractor(a, b,
iterations, x0, '', '', '', false);
33                     if (~isnan(attraction(1, end)) && ~isnan(
attraction(2, end)))
34                         if (any(ismembertol(attraction(1, :),
attractionDots(1, :), eps)) && any(ismembertol(attraction(2,
:), attractionDots(2, :), eps)))
35                             flag1(i, 1, p) = true;
36                             break;
37                         end
38                     end
39                 end
40                 [x0 outFlag3] = checkInitialPoint(a, b, FP, 2,
delta(p), false);
41                 if (outFlag3)
42                     %fprintf('%d2 x0 = (%.10f; %.10f)\n', p, x0
(1, 1), x0(1, 2));
43                     attraction = buildingHenonAttractor(a, b,
iterations, x0, '', '', '', false);
44                     if (~isnan(attraction(1, end)) && ~isnan(
attraction(2, end)))
45                         if (any(ismembertol(attraction(1, :),
attractionDots(1, :), eps)) && any(ismembertol(attraction(2,
:), attractionDots(2, :), eps)))
46                             flag1(i, 2, p) = true;
47                             break;

```

```

48         end
49     end
50 end
51 end
52 end
53 end
54 %disp(flag1);
55 out = false;
56 for i = 1:n
57     for j = 1:2
58         for l = 1:numberDelta
59             if (flag1(i, j, l) == true)
60                 out = true;
61             end
62         end
63     end
64 end
65 end

```

Листинг 8: **basinAttractionHenonSyst.m** — процедура построения бассейна притяжения для аттрактора.

```

1 function outArray = basinAttractionHenonSyst(a, b, xStart, xEnd,
    yStart, yEnd, stepSize, iterations, attractionDots, fileName,
    outDir, saveFlag)
2     disp('Starting creating basin...');
3     i = 1;
4     eps = 1e-6;
5     basin(1, 1) = 0; basin (2, 1) = 0;
6     PPs = henonPerpetualPoints(a, b);
7     [FP1 outFlag1] = henonFixedPoints(a, b, 1, false);
8     [FP2 outFlag2] = henonFixedPoints(a, b, 2, false);
9     n = 1;
10    for x = xStart:stepSize:xEnd
11        xCurr(n) = x;
12        n = n + 1;
13    end
14    m = 1;
15    for y = yStart:stepSize:yEnd
16        yCurr(m) = y;

```

```

17         m = m + 1;
18     end
19     n = n - 1; m = m - 1;
20     for o = 1:n
21         for p = 1:m
22             x0(1, 1) = xCurr(o);
23             x0(1, 2) = yCurr(p);
24             attraction = buildingHenonAttractor(a, b, iterations,
x0, '', '', '', '', false);
25             flag = false;
26             if (~isnan(attraction(1, end)) && ~isnan(attraction
(2, end)))
27                 if (any(ismembertol(attraction(1, :),
attractionDots(1, :), eps)) && any(ismembertol(attraction(2,
:), attractionDots(2, :), eps)))
28                     flag = true;
29                 end
30             end
31             if (flag)
32                 basin (1, i) = xCurr(o); basin(2, i) = yCurr(p);
33                 i = i + 1;
34             end
35         end
36     end
37     if (saveFlag)
38         h = figure ('Visible','off');
39         hold on;
40         color = [140, 140, 140] ./ 255;
41         pp = plot(basin(1, 1), basin(2, 1), 'w.', 'MarkerSize',
11);
42         fp = plot(basin(1, 1), basin(2, 1), 'w.', 'MarkerSize',
11);
43         basinDots = plot(basin(1, 1:end), basin(2, 1:end), 'k.',
'MarkerSize', 11, 'color', color);
44         if (imag(PPs(1, 1)) == 0) && (imag(PPs(1, 2)) == 0) && (~
isnan(PPs(1, 1))) && (~isnan(PPs(1, 2))) && (~isinfmy(PPs(1,
1))) && (~isinfmy(PPs(1, 2)))
45             pp = plot(PPs(1, 1), PPs(1, 2), 'g.', 'MarkerSize',
16);
46             text(PPs(1, 1), PPs(1, 2), ['PP1(' num2str(PPs(1, 1),

```

```

    '%.3g') ';' num2str(PPs(1, 2), '%.3g') ')]');
47     end
48     if (imag(PPs(2, 1)) == 0) && (imag(PPs(2, 2)) == 0) && (~
isnan(PPs(2, 1))) && (~isnan(PPs(2, 2))) && (~isinfmy(PPs(2,
1))) && (~isinfmy(PPs(2, 2)))
49         pp = plot(PPs(2, 1), PPs(2, 2), 'g.', 'MarkerSize',
16);
50         text(PPs(2, 1), PPs(2, 2), ['PP2(' num2str(PPs(2, 1),
%.3g') ';' num2str(PPs(2, 2), '%.3g') ')]');
51     end
52     if (outFlag1) && (~isinfmy(FP1(1))) && (~isinfmy(FP1(2)))
53         fp = plot(FP1(1), FP1(2), 'b.', 'MarkerSize', 16);
54         text(FP1(1), FP1(2), ['FP1(' num2str(FP1(1), '%.3g')
';' num2str(FP1(2), '%.3g') ')]');
55     end
56     if (outFlag2) && (~isinfmy(FP2(1))) && (~isinfmy(FP2(2)))
57         fp = plot(FP2(1), FP2(2), 'b.', 'MarkerSize', 16);
58         text(FP2(1), FP2(2), ['FP2(' num2str(FP2(1), '%.3g')
';' num2str(FP2(2), '%.3g') ')]');
59     end
60     line1 = sprintf('Henon basin with %.0f iterations',
iterations);
61     line2 = sprintf('a = %.5f, b = %.5f', a, b);
62     title({line1, line2});
63     legend([basinDots fp pp], 'Basin points', 'Fixed points',
'Perpetual points');
64     xlabel('x');
65     ylabel('y');
66     axis auto;
67     grid on;
68     set(h, 'Visible', 'on');
69     saveas(h,[outdir fileName], 'fig');
70     delete(h);
71 end
72 outArray = basin;
73 disp ('End creating basin');
74 end

```

Листинг 9: **vectorSyst.m** — вычисление новой пары параметров a и b для процедуры продолжения по параметру.

```

1 function out = vectorSyst(aCurr, aBegin, aEnd, bBegin, bEnd)
2 out = ((aEnd * bBegin - aBegin * bEnd) + (bEnd - bBegin) * aCurr)
   /(aEnd - aBegin);
3 end

```

Листинг 10: **isinfmy.m** — проверка, что точка находится в шаре радиуса 1000.

```

1 function out = isinfmy(S0)
2     flag = true;
3     if (size(S0, 2) == 2)&&(size(S0, 1) == 1)
4         if ((S0(1, 1)^2 + S0(1, 2)^2) < 1000)
5             flag = false;
6         end
7     end
8     if (size(S0, 1) == 2)&&(size(S0, 2) == 1)
9         if ((S0(1, 1)^2 + S0(2, 1)^2) < 1000)
10            flag = false;
11        end
12    end
13    if (size(S0, 1) == 1)&&(size(S0, 2) == 1)
14        if (S0 < 1000)
15            flag = false;
16        end
17    end
18    out = flag;
19 end

```


Приложение В

Реализация алгоритма построения карты динамических режимов и бифуркаций

Листинг 11: **dynamicMapModes.m** — процедура построения карты динамических режимов для системы Хенона.

```
1 function dynamicMapModes (aBegin, aEnd, bBegin, bEnd, aStep,
    bStep, q, k1, k2, iterations, outDir)
2     disp ('Starting Dynamic Map Modes...');
3     k = 1;
4     map(2, 1) = 0; map (1, 1) = 0;
5     for a = aBegin:aStep:aEnd
6         for b = bBegin:bStep:bEnd
7             map(1, k) = b; map(2, k) = a;
8             x0(1) = 0; x0(2) = 0;
9             attraction = buildingHenonAttractor(a, b, iterations,
    x0, '', '', '', '', false);
10            d = (attraction(1, end - q:end) + k1).^2 + (
    attraction(2, end - q:end) + k2).^2;
11            flag = false;
12            if ((isnan(d(q)) == 0)&&(isinf(d(q)) == 0)) && (~
    isinfmy(d(q)))
13                for i = q-1:-1:1
14                    if (d(q) == d(i))
15                        map(3, k) = q - i;
16                        flag = true;
17                        break;
18                    end
19                end
20                if (~flag)
21                    map(3, k) = 0;
22                end
23            else
24                map(3, k) = -1;
25            end
26            k = k + 1;
27        end
    end
```

```

28     end
29     h = figure ('Visible','off');
30     hold on;
31     dotsCycle1 = plot(map(1, 1), map(2, 1), 'w*', 'MarkerSize',
32 4);
33     dotsCycle2 = plot(map(1, 1), map(2, 1), 'w*', 'MarkerSize',
34 4);
35     dotsCycle3 = plot(map(1, 1), map(2, 1), 'w*', 'MarkerSize',
36 4);
37     dotsCycle4 = plot(map(1, 1), map(2, 1), 'w*', 'MarkerSize',
38 4);
39     dotsCycle5 = plot(map(1, 1), map(2, 1), 'w*', 'MarkerSize',
40 4);
41     dotsCycle6 = plot(map(1, 1), map(2, 1), 'w*', 'MarkerSize',
42 4);
43     dotsCycle7 = plot(map(1, 1), map(2, 1), 'w*', 'MarkerSize',
44 4);
45     dotsChaos = plot(map(1, 1), map(2, 1), 'w*', 'MarkerSize', 4)
46 ;
47     for i = 1:(k - 1)
48         if (map(3, i) == 1)
49             dotsCycle1 = plot(map(1, i), map(2, i), 'r*', '
50 MarkerSize', 4);
51         end
52         if (map(3, i) == 2)
53             dotsCycle2 = plot(map(1, i), map(2, i), 'y*', '
54 MarkerSize', 4);
55         end
56         if (map(3, i) == 3)
57             dotsCycle3 = plot(map(1, i), map(2, i), 'b*', '
58 MarkerSize', 4);
59         end
60         if (map(3, i) == 4)
61             dotsCycle4 = plot(map(1, i), map(2, i), 'c*', '
62 MarkerSize', 4);
63         end
64         if (map(3, i) == 5)
65             dotsCycle5 = plot(map(1, i), map(2, i), 'g*', '
66 MarkerSize', 4);
67         end

```

```

55         if (map(3, i) == 6)
56             dotsCycle6 = plot(map(1, i), map(2, i), 'm*', '
MarkerSize', 4);
57         end
58         color = [251, 133, 24] ./ 255;
59         if (map(3, i) == 7)
60             dotsCycle7 = plot(map(1, i), map(2, i), '*', '
MarkerSize', 4, 'color', color);
61         end
62         if (map(3, i) == 0)
63             dotsChaos = plot(map(1, i), map(2, i), 'k*', '
MarkerSize', 4);
64         end
65     end
66     line1 = sprintf('Dynamic map modes with %.0f iterations',
iterations);
67     title({line1});
68     legend([dotsCycle1, dotsCycle2, dotsCycle3, dotsCycle4,
dotsCycle5, dotsCycle6, dotsCycle7, dotsChaos], 'Points cycle
period 1', 'Points cycle period 2', 'Points cycle period 3', '
Points cycle period 4', 'Points cycle period 5', 'Points cycle
period 6', 'Points cycle period 7', 'Points chaos');
69     xlabel('b');
70     ylabel('a');
71     axis auto;
72     grid on;
73     set(h, 'Visible', 'on');
74     saveas(h,[outDir '/' Map (' num2str(aBegin, '%.6g') ';' num2str
(bBegin, '%.6g') ') to (' num2str(aEnd, '%.6g') ';' num2str(
bEnd, '%.6g') ').fig'], 'fig');
75     delete (h);
76     disp ('End Dynamic Map Modes');
77 end

```

Листинг 12: **bifurcationHenonSyst.m** — процедура построения бифуркационных диаграмм для системы Хенона.

```

1 function bifurcationHenonSyst(aBegin, aEnd, bBegin, bEnd, aStep,
    bStep, iterations, saveFlag, ppFpFlag, outDir)
2     disp ('Starting bifurcations...');

```

```

3     h = figure ('Visible', 'off');
4     hold on;
5     pp = plot(0, 0, 'w.', 'MarkerSize', 10);
6     fp = plot(0, 0, 'w.', 'MarkerSize', 10);
7     for alpha = aBegin:aStep:(aEnd)
8         x0(1) = 0; x0(2) = 0;
9         attraction = buildingHenonAttractor(alpha, bEnd,
10 iterations, x0, '', '', '', false);
11         bifurcationDots = plot(attraction(1, iterations - 9:
12 iterations), alpha * ones(1,10), '.k', 'MarkerSize', 8);
13         if (ppFpFlag)
14             PPs = henonPerpetualPoints(alpha, bEnd);
15             [FP1 outFlag1] = henonFixedPoints(alpha, bEnd, 1,
16 false);
17             [FP2 outFlag2] = henonFixedPoints(alpha, bEnd, 2,
18 false);
19             if (imag(PPs(1, 1)) == 0) && (imag(PPs(1, 2)) == 0)
20 && (~isnan(PPs(1, 1))) && (~isnan(PPs(1, 2))) && (~isinfmy(PPs
21 (1, 1))) && (~isinfmy(PPs(1, 2)))
22                 pp = plot(PPs(1, 1), PPs(1, 2), 'g.', 'MarkerSize
23 ', 10);
24             end
25             if (imag(PPs(2, 1)) == 0) && (imag(PPs(2, 2)) == 0)
26 && (~isnan(PPs(2, 1))) && (~isnan(PPs(2, 2))) && (~isinfmy(PPs
27 (2, 1))) && (~isinfmy(PPs(2, 2)))
28                 pp = plot(PPs(2, 1), PPs(2, 2), 'g.', 'MarkerSize
29 ', 10);
30             end
31             if (outFlag1) && (~isinfmy(FP1(1))) && (~isinfmy(FP1
32 (2)))
33                 fp = plot(FP1(1), FP1(2), 'b.', 'MarkerSize', 10)
34 ;
35             end
36             if (outFlag2) && (~isinfmy(FP2(1))) && (~isinfmy(FP2
37 (2)))
38                 fp = plot(FP2(1), FP2(2), 'b.', 'MarkerSize', 10)
39 ;
40             end
41         end
42     end
43 end

```

```

29     line1 = sprintf('Bifurcation diagram for the Henon Map with
%.0f iterations', iterations);
30     line2 = sprintf('aBegin = %.5f to aEnd = %.5f', aBegin, aEnd)
;
31     title({line1, line2});
32     if (ppFpFlag)
33         legend([bifurcationDots fp pp], 'Bifurcation dots', '
Fixed points', 'Perpetual points');
34     else
35         legend(bifurcationDots, 'Bifurcation dots');
36     end
37     xlabel ('x');
38     ylabel ('a');
39     axis auto;
40     grid on;
41     if (saveFlag)
42         set(h, 'Visible', 'on');
43         saveas(h,[outdir '/Bifurcations for parameters aBegin = '
num2str(aBegin,'%0.6g') ' to aEnd = ' num2str(aEnd,'%0.6g') '.
fig'], 'fig');
44     end
45     delete(h);
46     delete(bifurcationDots);
47
48     h = figure ('Visible', 'off');
49     hold on
50     pp = plot(0, 0, 'w.', 'MarkerSize', 10);
51     fp = plot(0, 0, 'w.', 'MarkerSize', 10);
52     for beta = bBegin:bStep:(bEnd)
53         x0(1) = 0; x0(2) = 0;
54         attraction = buildingHenonAttractor(aEnd, beta,
iterations, x0, '', '', '', false);
55         bifurcationDots = plot(attraction(1,iterations - 9:
iterations), beta * ones(1,10), '.k', 'MarkerSize', 8);
56         if (ppFpFlag)
57             PPs = henonPerpetualPoints(aEnd, beta);
58             [FP1 outFlag1] = henonFixedPoints(aEnd, beta, 1,
false);
59             [FP2 outFlag2] = henonFixedPoints(aEnd, beta, 2,
false);

```

```

60         if (imag(PPs(1, 1)) == 0) && (imag(PPs(1, 2)) == 0)
&& (~isnan(PPs(1, 1))) && (~isnan(PPs(1, 2))) && (~isinfmy(PPs
(1, 1))) && (~isinfmy(PPs(1, 2)))
61             pp = plot(PPs(1, 1), PPs(1, 2), 'g.', 'MarkerSize
', 10);
62         end
63         if (imag(PPs(2, 1)) == 0) && (imag(PPs(2, 2)) == 0)
&& (~isnan(PPs(2, 1))) && (~isnan(PPs(2, 2))) && (~isinfmy(PPs
(2, 1))) && (~isinfmy(PPs(2, 2)))
64             pp = plot(PPs(2, 1), PPs(2, 2), 'g.', 'MarkerSize
', 10);
65         end
66         if (outFlag1) && (~isinfmy(FP1(1))) && (~isinfmy(FP1
(2)))
67             fp = plot(FP1(1), FP1(2), 'b.', 'MarkerSize', 10)
;
68         end
69         if (outFlag2) && (~isinfmy(FP2(1))) && (~isinfmy(FP2
(2)))
70             fp = plot(FP2(1), FP2(2), 'b.', 'MarkerSize', 10)
;
71         end
72     end
73 end
74 line1 = sprintf('Bifurcation diagram for the Henon Map with
%.0f iterations', iterations);
75 line2 = sprintf('bBegin = %.5f to bEnd = %.5f', bBegin, bEnd)
;
76 title({line1, line2});
77 if (ppFpFlag)
78     legend([bifurcationDots fp pp], 'Bifurcation dots', '
Fixed points', 'Perpetual points');
79 else
80     legend(bifurcationDots, 'Bifurcation dots');
81 end
82 xlabel('x');
83 ylabel('b');
84 axis auto;
85 grid on;
86 if (saveFlag)

```

```

87         set(h, 'Visible', 'on');
88         saveas(h,[outDir '/Bifurcations for parameters bBegin = '
            num2str(bBegin,'%0.6g') ' to bEnd = ' num2str(bEnd,'%0.6g') '.
fig'], 'fig');
89     end
90     delete(h);
91     disp ('End bifurcations');
92 end

```

Листинг 13: **runHenonAnalysis.m** — запуск построения бифуркационных диаграмм и карты динамических режимов.

```

1 function runHenonAnalysis
2     clc
3     outDirBifurcation = './OUT/Bifurcations';
4     outDirDynamicMapModes = './OUT/Dynamic map modes';
5     outDirBas = './OUT/Basin';
6     if ~exist(outDirBifurcation, 'dir')
7         mkdir(outDirBifurcation);
8     end
9     if ~exist(outDirDynamicMapModes, 'dir')
10        mkdir(outDirDynamicMapModes);
11    end
12    if ~exist(outDirBas, 'dir')
13        mkdir(outDirBas);
14    end
15
16    aBegin = 0.1; aEnd = 1.502;
17    bBegin = -0.6; bEnd = -0.138;
18    aStep = 0.001; bStep = 0.001;
19    iterations = 1000;
20    saveFlag = true;    ppFpFlag= false;
21    bifurcationHenonSyst(aBegin, aEnd, bBegin, bEnd, aStep, bStep
, iterations, saveFlag, ppFpFlag, outDirBifurcation);
22
23    a = 1.4; b = 0.3;
24    xStart = -3; xEnd = 3;
25    yStart = -3; yEnd = 3;
26    x0(1) = 0; x0(2) = 0;
27    stepSize = 0.01;

```

```

28     iterations = 1000;
29     saveFlag = true;
30     name = ['/Attractor (' num2str(a, '%.6g') ';' num2str(b, '%.6g'
) '.fig'];
31     line1 = sprintf('Henon attractor with %.0f iterations',
iterations);
32     line2 = sprintf('a = %.5f, b = %.5f', a, b);
33     color = [0 0 0]./255;
34     inTitle = {line1, line2};
35     attractorDots = buildingHenonAttractor(a, b, iterations, x0,
name, outDirBas, inTitle, color, true);
36     name = ['/Basin for parameter (' num2str(a, '%.6g') ';'
num2str(b, '%.6g') ').fig'];
37     basinAttractionHenonSyst(a, b, xStart, xEnd, yStart, yEnd,
stepSize, iterations, attractorDots, name, outDirBas, saveFlag
);

38
39     aBegin = -3; aEnd = 3;
40     bBegin = -2; bEnd = 2;
41     aStep = 0.01; bStep = 0.01;
42     %aBegin = 1.49-0.3; aEnd = 1.49+0.3;
43     %bBegin = -0.138-0.1; bEnd = -0.138+0.1;
44     %aStep = 0.001; bStep = 0.001;
45     %-----
46     q = 8;
47     k1 = 2; k2 = 4;
48     iterations = 1000;
49     dynamicMapModes(aBegin, aEnd, bBegin, bEnd, aStep, bStep, q,
k1, k2, iterations, outDirDynamicMapModes);
50 end

```